

In [41]:

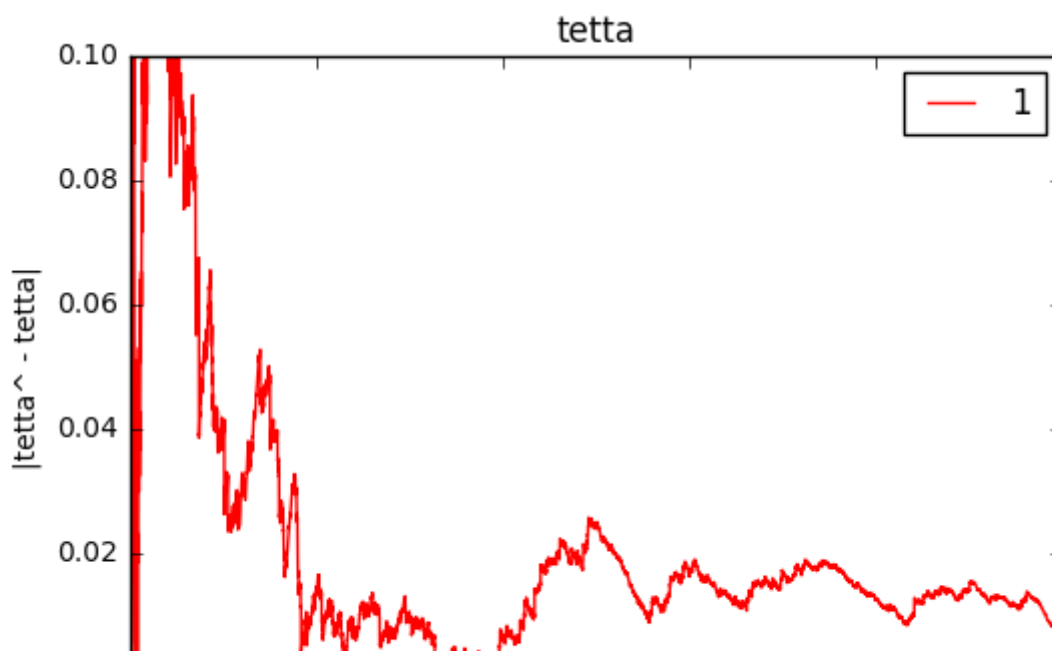
```
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from pylab import *
def fac(n):
    if n == 0:
        return 1
    return fac(n-1) * n
```

In [42]:

```
N = 10000
tetta = 1.0
# Выборка из экспоненциального распределения с параметром tetta(параметр фиксирован
s_all = np.random.exponential(tetta, N)
```

In [45]:

```
# Для различных параметров k строятся графики зависимости модуля разности оценки с
for k in range(2, 13, 1):
    s = s_all ** k
    arr = []
    for n in range(1, N + 1):
        arr += [abs((fac(k) / (s[:n].sum() / n)) ** (1.0 / k) - tetta)]
    ylim(0,0.1)
    plt.plot(range(1, N + 1), arr, color = 'red', label='1')
    plt.legend(loc='best')
    plt.title('tetta')
    plt.xlabel('n')
    plt.ylabel('|tetta^k - tetta|');
    plt.show()
```



Вывод

При фиксированной изначально выборке с увеличением k оценка становилась хуже в смысле модуля разности с истинным значением